

Part1, Particles in a box

1 Particles in a box

To be able to send a satellite out into the vastness of space we will need some sort of fuel for takeoff and also for correctional boosts whilst in space to make sure we reach our destination. In this part we will create a program which calculates the amount of fuel (in kg) we will need for each required boost. But first a small introduction to rocket science and how to apply it to our project!

In short: This part of the project is very similar to exercises 1A.6 and 1A.7 in lecture notes part 1A. It is recommended to do these two exercises first, then write the slightly more general routine that you need for the project. The goal of this part of the project is simple: you need to write a code which calculates the amount of fuel you need for a given boost Δv . This is exactly what you do in exercise 1A.7, but here you need to make the code valid for any boost Δv , not just to reach escape velocity. During your trip to another planet, your satellite will need to do several correctional maneuvers, and each of these will need fuel. You need the code that you write here later in the project when you have made a planned trajectory with a planned set of boosts so that you can calculate the total amount of fuel you need to bring before launching your rocket.

Note: The codes you write in this part will only be used by yourself (or within your two-person group) and will probably not be called from other codes during the project. You will probably need to run this code again at a later stage in the project in order to adjust your fuel calculations, but apart from that, all you need from this part for the rest of the project are the numbers that you get from running the code.

As you might have read (if not, go back and read it) in **Lecture Notes Part 1A** a rocket engine consists of a fuel tank and an ignition chamber where there are millions and millions of gas particles stacked together. The purpose of the fuel tank is mainly to supply the ignition chamber with fuel (gas particles). Inside the ignition chamber the combustion of the gas creates massive temperatures which in turn creates a massive pressure inside the compartment. The exhaust particles takes part in elastic collisions with the walls of the chamber (remember from FYS-MEK1100, elastic collisions preserve both momentum and mechanical energy). Since the collisions are elastic the particles do not lose energy or momentum inside the compartment. If we now open a hole at the bottom of the chamber (the one facing the ground) particles will leave the compartment through this hole, creating a net momentum loss Δp towards the ground. In order for total momentum to be conserved, the fuel tank (and indeed the rocket/satellite) must gain momentum $-\Delta p$ in the opposite direction.

A rocket therefore gets its thrust from the momentum of the exhaust particles leaving the bottom of the rocket. These particles have momentum in the direction pointing towards the ground which gives the rockets momentum up and away from the ground.

This is the general simplified idea of a rocket engine. In this project we will make some shortcuts and assumptions to make sure the project isn't overwhelming but at the same time preserving as much physics as possible.

We will make the following assumptions and short cuts:

1. The density and temperature and thereby the pressure inside the ignition chamber is constant throughout the boost (this means that even though the amount of particles inside the ignition chamber would normally decline towards the end of the boost, we keep it constant).
2. Although the particles escaping would normally be some sort of exhaust particles we will use H_2 gas.
3. The gas does not ignite (we will simply heat up the gas towards the required temperatures, but the particles inside the fuel compartment are the same that leave the nozzle [exit hole]).
4. In order to simulate the rocket engine and calculate the fuel consumption for a given boost, you will, as specified in exercise 1A.7, simulate the boost during a finite time interval (often of a few minutes). However for the actual boost you give the rocket through `AST2000SolarSystem` later on will, for simplicity, be treated as instant. More information about the boost will follow later in the assignment.
5. The particles do not interact/collide with each other inside the compartment. Had they done so this would still not effect the total energy of the tank since these collisions would be elastic.

1.1 Warming up and testing your understanding

We will start by working with the main concepts of part 1A of the lecture notes to make sure you have the necessary understanding to make the rocket engine, but also to calculate some analytical expressions which you may need to solve the numerical part. Most of this warming-up exercise is optional for those working on the project alone: It is recommended that you do it (for your understanding), but you strictly only need to do the first two points (plotting the distribution function) as well as exercise 1A5.

1. Start by reading through exercise 1A.6: what is the temperature and density of the gas in your rocket?
2. What is the statistical distribution function you need to find the distribution of velocities along one given dimension in your gas? Make a plot with v_x on the x -axis and probability density on the y -axis. Use a range from -25000m/s to 25000m/s on the x -axis.
3. Use the plot to make a rough by-eye estimate of the number of hydrogen molecules per m^3 in the rocket engine which have a velocity in the

range $[5000\text{m/s}, 30000\text{m/s}]$. Then use your favourite numerical integration method to calculate the exact number.

4. Now make the same calculations, by-eye and numerically to find the number of hydrogen atoms in the velocity range $[-8000\text{m/s}, 8000\text{m/s}]$.
5. Now do exercise 1A3 followed by 1A5.
6. What is the mean kinetic energy of a particle in the rocket engine? And the mean velocity?

1.2 Simulating particles in a box

To make sure you have a good understanding of how these particles behave, your first assignment will be to create and simulate the particles inside a box. These particles will need velocity distributions as explained in the lecture notes. Make sure that the particles behave as predicted in your simulation before continuing calculating the amount of fuel you need. Included in the code skeleton is a plotting example which you can use to animate the particles in your box in order to check that they behave as expected. To prevent the animation from being too slow, use very few particles (maybe about 100). When you see that the movement of the particles seems correct, you can continue coding, increasing the number of particles to at least 100000 as explained in exercise 1A.6.

1.3 Calculating the needed fuel amount

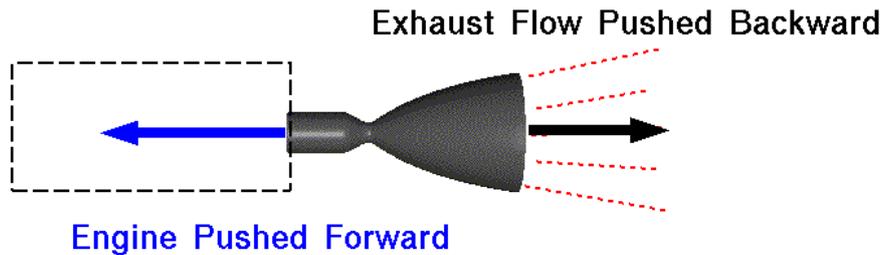
We are now ready to calculate how much fuel we will need for a given boost. First of all you will have to create a hole in your box through which the particles can escape and therefore accelerating the rocket/satellite. You will have to count the particles escaping through this hole and their momentum in the direction normal to the surface area of the hole.



Newton's Third Law



Rocket Engine Thrust



For every action, there is an equal and opposite re-action.

Figure 1: A NASA figure of the fundamentals of rocket engines

Remember one of our assumptions earlier in part1: the density and pressure inside the compartment is constant. The amount of particles that escape the hole as well as their momentum is therefore also constant causing the change in momentum of our rocket engine to be constant. The force in the boost-direction is given by $F = \frac{dp}{dt}$. A crucial point here is that the particles behave the same way inside a small compartment as they do inside a big compartment. This means you can (to make the simulations faster) create a small fuel compartment and find out how much acceleration it gives you in the boost direction. You can then just combine multiple identical small compartments (the equivalent of having 1 big one) until you get the required acceleration.

You have now (hopefully) grasped how to accelerate your rocket. What remains now is how much fuel you will need.

The goal of this part is twofold. First you need a code for the launch sequence. In exercise 1A.7 you calculate the amount of fuel that you need in order to reach escape velocity in 20 minutes, you need to write a code which does this for your home planet (planet 0) in your random solar system. Second, you need to write a code to calculate the amount of fuel needed to give the rocket a given boost Δv in space (where gravity is negligible). To do this you will have to split up your boost into small time intervals. Here you need to experiment to find a way to calculate the time and the total fuel needed for a general velocity change Δv . You may adjust gas density, temperature, total boost time and the size of the hole in order to obtain your goal. The less fuel and the less acceleration time you manage to get, the better. But it is difficult to get both these measures low at the same time. You will need to find a compromise.

In the end you should be able to calculate how many kilograms of fuel you need to boost a rocket with mass 1100 kg.

For the proper launch to reach escape velocity, you will need to include the mass of all your fuel in addition to the mass of the rocket. Remember to reduce the total mass of the rocket for each small boost in time step dt .

Testing Your Code

At some point you would like to check if your particle simulation and launch (until escape velocity) is reached. This has to be done to proceed with part 2.

The test consists in launching your rocket at time $T = 0$, from the surface of your home planet. What you need to calculate is the exact position vector of the rocket (relative to the star, in units of AU), at the point in time when *your engine* has lifted and accelerated the rocket such that *the kinetic energy is equal to the potential energy* from your home planet (escape velocity).

To do this, you will need to load your personal solar system (like shown in the general introduction), because in addition to your launch height before reaching escape velocity, you need to include both the effect of your planet having a velocity around the sun, and the planet surface having a velocity because of self-rotation. These two effects combined means that the rocket will have a non-zero initial velocity *relative to the sun*, which needs to be included in the launch.

After calculating the x and y position at the end of the launch, you can check if it is correct by first setting up your engine, then running `mass_needed_launch` with `test=True`.

```
1 syst = AST2000SolarSystem(seed)
2
3 #satellite on surface in positive x-direction
4 init_sat_pos = (x0[0] + radius.in_AU, y0[0])
5
6 #set up the engine with parameters from your simulation
7 syst.engine_settings(dpd_t1box, no_boxes,
8 particles_sec1box, initial_fuel_mass,
9 T_launch, init_sat_pos, 0) #0 means launch time = 0
10
11 pos_after_launch = (calculated_x, calculated_y)
12
13 syst.mass_needed_launch(pos_after_launch, test=True)
```

More Hints

On the course webpage, there is a skeleton python code, `part1_skeleton.py`, that will help you a lot in completing this simulation. We are in fact encouraging you to use this code skeleton to refresh your knowledge of vectorization and loops, which you will need to use effectively to complete the rest of the project.

The temperature of burning hydrogen is about 3000 K, but because the rest of the engine is not completely accurate, we will allow temperatures up to 10 000 K.

The rest of the project is even more difficult than this first part. If you struggle with completing part 1 in one week, don't hesitate to contact the group teacher for advice.